International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 6 Issue 6, September-October 2022 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Novel Image Compression Approach-Inexact Computing

Sonam Kumari[1], Manish Rai[2]

[1]Student, [2]Assistant Professor,

[1,2]Bhabha Engineering Research Institute, Bhopal, Madhya Pradesh, India

ABSTRACT

This work proposes a novel approach for digital image processing that relies on faulty computation to address some of the issues with discrete cosine transformation (DCT) compression. The proposed system has three processing stages: the first employs approximated DCT for picture compression to eliminate all compute demanding floating-point multiplication and to execute DCT processing with integer additions and, in certain cases, logical right / left modifications. The second level reduces the amount of data that must be processed (from the first level) by removing frequencies that cannot be perceived by human senses. Finally, in order to reduce power consumption and delay, the third stage employs erroneous circuit level adders for DCT computation. A collection of structured pictures is compressed for measurement using the suggested three-level method. Various figures of merit (such as energy consumption, delay, power-signal-to-noise-ratio, average-difference, and absolute-maximum-difference) are compared to current compression techniques; an error analysis is also carried out to substantiate the simulation findings. The results indicate significant gains in energy and time reduction while retaining acceptable accuracy levels for image processing applications.

KEYWORDS: *Approximate computing, DCT, inexact computing, image compression*

How to cite this paper: Sonam Kumari | Manish Rai "A Novel Image Compression Approach-Inexact Computing" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-6 | Issue-6, October 2022, pp.1988-1994, URL: www.ijtsrd.com/papers/ijtsrd52197.pdf

IJTSRD52197

1. INTRODUCTION:

TODAY'S amount of information that is computational and power computing system usually process a significant intensive. Digital Signal Processing (DSP) systems are widely used to process image and video information, often under mobile/wireless environments. These DSP systems use image/video compression methods and algorithms. However, the demands of power and performance remain very stringent. Compression Methods are often used to ease such needs. Image/video compression techniques are classified into two types: lossless and lossy.

The latter type is more hardware efficient, but at the sacrifice of ultimate decompressed image/video quality. The Joint Photographic Experts Group (JPEG) technique is the most extensively used lossy approach for image processing, while the Moving Picture Experts Group (MPEG) method is the most widely used lossy method for video processing. As the first processing stage, both standards use the Discrete Cosine Transform (DCT) algorithm. Many other rapid DCT [1], [2] computing techniques have been developed for picture and video applications; however, since all of these algorithms still use floating point multiplications, they are computationally demanding and require substantial hardware resources. To solve these problems, many algorithms, such as [3], may have their coefficients scaled and approximated by integers, allowing floating-point multiplications to be substituted by integer multiplications [4], [5].

Because the resultant algorithms are substantially quicker than the original ones, they are widely employed in practical applications. As a result, the design of excellent DCT approximations for implementation by lower bus width and simpler arithmetic operations (such as shift and addition) has gained a lot of attention in recent years [6]. Image/video processing has the benefit of being extremely error-tolerant; human senses cannot typically detect decrease in performance, such as visual and audio information quality. As a result,

imprecise computing may be employed in many applications that allow some loss of accuracy and uncertainty, such as image/video processing [7, 8]. The introduction of inaccuracy at the circuit level in the DCT calculation is difficult and targets certain figures of merit (such as power dissipation, latency, and circuit complexity [9], [10], [11], [12], [13], [14]). This method is aimed towards low-power users. A logic/gate/transistor level redesign of an exact circuit is used to achieve process tolerance. A logic synthesis technique [9] has been described for designing circuits for implementing an inexact version of a given function using error rate (ER) as a parameter for error tolerance. Reducing the complexity of an adder circuit at the transistor level (for example, by truncating the circuits at the lowest bit positions) reduces power dissipation more than conventional low power design techniques [10]; in addition to the ER, new figures of merit for estimating the error in an inexact adder have been presented in [11]. This study introduces a novel framework for approximation DCT image compression, which is based on inexact computation and has three layers. Level 1 is a multiplier-less DCT transformation that uses just adds; Level 2 is high frequency component (coefficient) filtering; and Level 3 is computation utilizing inexact adders. Level 1 has received much attention in the technical literature [16], [17], and [18]; Level 2 is an obvious strategy for reducing computing complexity while achieving just a minimal loss in picture compression. Level 3 employs a circuit level method to pursue inexact computation (albeit new and efficient inexact adder cells are utilized in this manuscript). As a result, the significance of this text may be found in the combined consequences of these three levels. The suggested framework has been thoroughly examined and appraised. For picture compression as an application of inexact computing, simulation and error analysis demonstrate remarkable consistency in findings. To prevent misunderstanding, the term "approximate" refers specifically to DCT methods, while the term "inexact" refers to circuits and designs that employ non-exact hardware to compute the DCT.

2. REVIEW OF DCT For manuscript completeness, preliminaries to approximate DCT and a review of relevant topics are presented next.

2.1. Discrete Cosine Transform (DCT) To obtain the ith and jth DCT transformed elements of an image block (represented by a matrix p of size N), the following equation is used:

$$D(i,j) = \frac{1}{\sqrt{2N}}C(i)C(j)$$

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(x,y)\cos\left[\frac{\pi(2x+1)i}{2N}\right]\cos\left[\frac{\pi(2y+1)j}{2N}\right] \quad (1)$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, u = 0 \\ 1, u > 0 \end{cases}$$

Where $p(x,y)$ is the x,yt element of the image. This equation calculates one entry (i,jt) of the

Transformed image from the pixel values of the original image matrix. For the commonly used 8×8

Block for JPEG compression, N is equal to 8 and x *and* y from 0 *to* 7. Therefore $D(I,j)$ is also given

By the following equation:

$$D(i,j) = \frac{1}{4}C(i)C(j)$$

$$\sum_{i=0}^{7}\sum_{j=0}^{7} p(x,y)\cos\left[\frac{\pi(2x+1)i}{16}\right]\cos\left[\frac{\pi(2y+1)j}{16}\right] \quad (2)$$

For matrix calculations, the SCT matrix is obtained from the following:

$$T_{DCT}(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, i = 0 \\ \sqrt{\frac{2}{N}}\cos\left[\frac{\pi(2j+1)i}{2N}\right], i > 0 \end{cases} \quad (3)$$

So, DCT is computation intensive and may require floating-point operations for processing, Unless an approximate algorithm is utilized.

2.2. Joint Photographic Experts Group (JPEG) The JPEG processing is first initiated by transforming an image to the frequency domain using the DCT; this separates images into parts of differing frequencies. Then, the quantization is performed such that frequencies of lesser importance are discarded. This reflects the capability of humans to be reasonably good at seeing small differences in brightness over a relatively large area, but they The precise degree of a quickly fluctuating brightness change is frequently indistinguishable. During this quantization stage, each frequency domain component is split by a constant and then rounded to the closest integer to compress it. As a consequence, many high frequency components have extremely tiny or probable zero values, at best negligible values. The picture is then recovered during the decompression process, which is carried out with just the necessary frequencies kept. The following actions must be taken before JPEG processing can begin:

1. An picture (in color or grayscale) is first segmented into kxk pixel blocks (k = 8 is typical).
2. The DCT is then applied to each block from left to right and top to bottom.
3. This produces kxk coefficients (so 64 for k = 8) which are quantized to minimize the magnitudes.
4. The compressed picture, i.e., the stored or communicated image, is represented by the resultant array of compressed blocks. 5. To obtain the picture, decompress the compressed image (array of blocks) using Inverse DCT (IDCT).

3. INEXACT ADDITION AND APPROXIMATE DCT

Arithmetic circuits are particularly adapted to inexact computation; also, they have been widely studied in the technical literature.

TABLE 1
Approximate DCT Methods Applied to Image Compression; Number of Operations Required to Calculate the DCT for an 8x8 Block Size

	Method	Additions	Multiplications	Shifts	Total operations
Multipliers	DFT by definition [1]	56^a (432)	64^b (192)	0	624
	DFT, Cooley–Tukey [1]	24^a (58)	2^b (6)	0	64
	DCT by definition [2]	56	64	0	120
	Arai algorithm [3]	29	5	0	34
Multiplier-less	SDCT [23]	24	0	0	24
	BAS08 [25]	18	0	2	20
	BAS09 [26]	18	0	0	18
	BAS11 [27] with $a = 0$	16	0	0	16
	BAS11 [27] with $a = 1$	18	0	0	18
	BAS11 [27] with $a = 2$	18	0	2	20
	CB11 [28]	22	0	0	22
	BC12 [29]	14	0	0	14
	PEA12 [16]	24	0	6	30
	PEA14 [17]	14	0	0	14

Literature and is an essential arithmetic operation in many inexact computer applications. A decrease in circuit complexity at the transistor level of an adder circuit frequently results in a significant reduction in power dissipation, which is often more than that provided by standard low power design approaches [10]. In [12], inexact adder designs were evaluated: inexact operation was introduced by either replacing the accurate cell of a modular adder design with a reduced circuit complexity approximation cell, or by changing the production and propagation of the carry in the addition process. Three novel inexact adder cell designs (denoted as InXA1, InXA2, and InXA3) are reported in [14]; these cells have both electrical and error properties that are particularly advantageous for approximation computation. These adder cells, as shown in Table 1, offer the following advantages over prior designs [10], [13]: I a limited number of transistors; (ii) a small number of erroneous outputs at the two outputs (Sum and Carry); and (iii) reduced switching capacitances (represented in Cgn gate capacitance of minimal size NMOS), resulting in a significant decrease in both delay and energy dissipation (Table 3). (and their product as combined metric). Table 1 displays metrics such as latency, energy wasted, and EDP (energy delay product) of inexact cells for both average and worst instances. InXA1 is the least average and lowest performing inexact cell. case InXA2 incurs delays in the least average and worst case power dissipations, as well as the least average EDP. Extensive modeling was used to establish the average and worst-case latency and energy dissipation of the adder cells. The delay for each input signal is recorded when the output reaches 90% of its maximum value, whereas the energy wasted in all transistors is measured when the output reaches 90%. Based on these benefits, InXA1 and InXA2 based adders are being explored for the DCT application, which will be discussed more below.

4. APPROXIMATE FRAMEWORKS PROPOSED

This research introduces a novel picture compression framework with three layers of approximation, as seen below.

Level 1 represents the multiplier-less DCT transformation, Level 2 represents high frequency filtering, and Level 3 represents inexact calculation. Levels 1 and 3 have already been discussed. Although high frequency filtering (Level 2) is not a novel idea, it is worth describing it for completeness' sake since it adds to the proposed framework's execution time and energy savings. As a consequence, rather than executing the quantization process on all resultant DCT transformation coefficients, the operation is only conducted on the set of coefficients for the changed block's low frequency components.

4.1. High Frequency Filtration

Filtering the high frequencies results in a picture that is hardly discernible by the human eye (as only sensitive to low frequency contents).

This functionality allows you to compress a picture. As previously stated, a DCT changes the picture in the frequency domain such that the coefficients that represent the high frequency components (and so are not visible to the human eye) may be ignored while the other coefficients are retained. When used to picture compression applications, different amounts of retained coefficients are investigated; it has been proved that just 0.34-24.26 percent of 92112 DCT coefficients are adequate in high speed face recognition applications.

Image compression using a supporting vector machine that only considers the top 8-16 coefficients, As suggested an image reconstruction approach based only on three coefficients. Evaluation and comparison of several picture compression algorithms using just ten coefficients, as described in [25].

4.2. DCT Implementation Estimate

- Unlike the approximate DCT techniques shown in Table 1, all needed computations (addition and subtraction) are then executed at the bit level using the associated logic functions. The length of all operators is 32 bits, and MATLAB simulates implementations using their Boolean logical functions.
- Selected approximation DCT techniques are simulated for the Lena picture, and the results are presented in Fig. 1, where the Power Signal to Noise Ratio (PSNR) of all methods is plotted against the number of Retained Coefficients (RC) utilized in the compression's quantization step. The PSNR is derived as follows from the Mean Square Error (MSE):Mean Square Error (MSE):

$$MSE = \frac{1}{m \times n} \sum_{x=1}^{m} \sum_{y=1}^{n} (p_{x,y} - \hat{p}_{x,y})^2 \quad (5)$$

- Where p (j,k) is the image's correct pixel value at row x and column y, p (x,y) is the approximation value of the same pixel, and m and n are the image's dimensions (rows and columns respectively). Peak Signal to Noise Ratio (PSNR):

$$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} \quad (6)$$

Except for the non-orthogonal SDCT approach, the data reveal that compression utilizing CB11 delivers the best PSNR values. There are three categories of behavior noticed. Increasing output quality as the number of retained increases coefficients (RC). This By raising the RC for CB11, BAS08, BAS09, and BAS11 (a = 0 and a = 1), a virtually constant PSNR is obtained. This happens with BC12 and PEA14, resulting in a decrease in output quality as the RC increases. This happens with both BAS11 (a = 2) and PEA12. Two additional metrics, the Average Difference (AD) and the Maximum Absolute Difference (MAD), are employed to get a better understanding of the resultant quality (MD). These metrics are defined as follows: Average Difference (AD):

$$AD = \frac{1}{m \times n} \sum_{j=1}^{m} \sum_{k=1}^{n} (p_{x,y} - \hat{p}_{x,y}) \quad (7)$$

- Maximum Absolute Difference (MD):

$$MD = \max_{m,n} \{ |p_{x,y} - \hat{p}_{x,y}| \} \quad (8)$$

Figs. 2 shows the resulting AD and MD for all methods; the average difference between the uncompressed and inexact-compressed images become smaller as RC increases except for BAS11(a =2) and PEA12 (further confirming the PSNR results

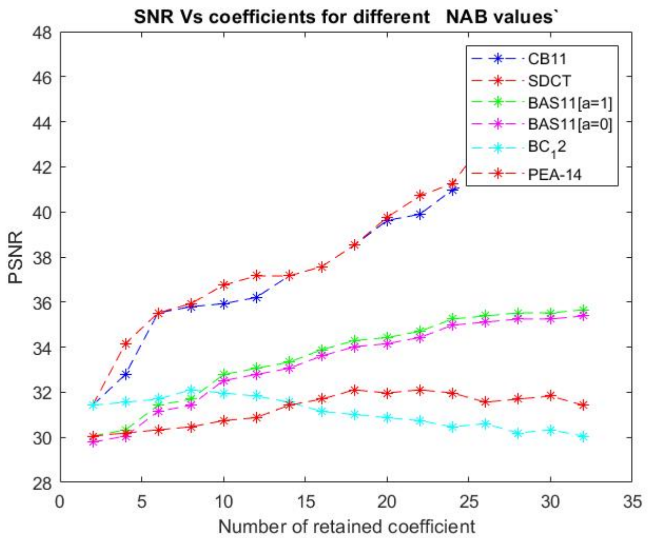

Fig1: Compression of an image using approximate DCT and bit-level exact computing.

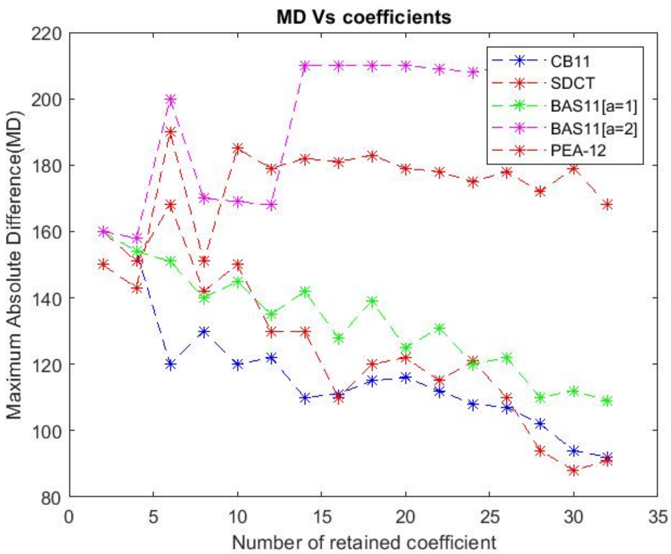

Fig2: Maximum absolute difference (MD) for compression of an image using approximate DCT

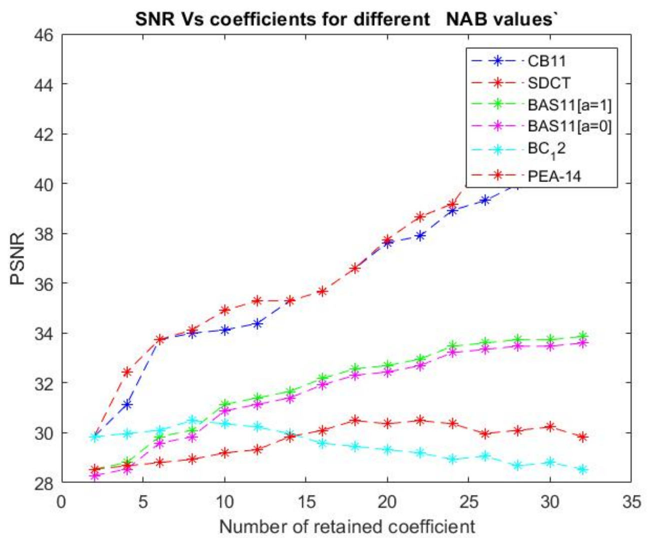

Fig3: Approximate DCT compression of an image using inexact adders with different NAB values; (Number of Approximate bits) NAB =3

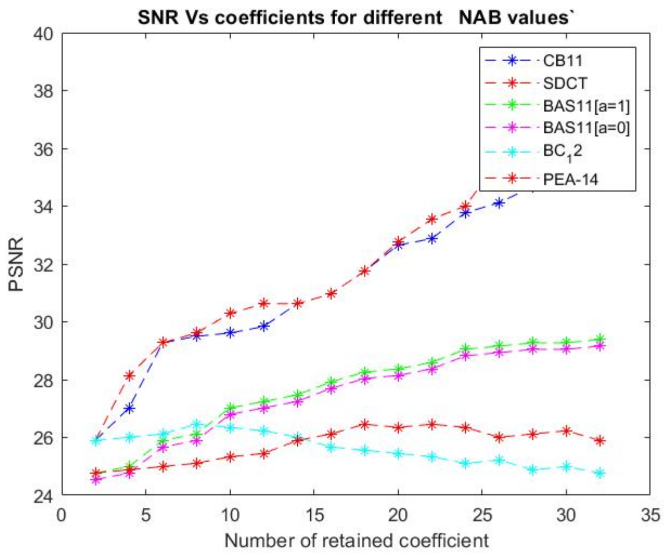

Fig4: Approximate DCT compression of an image using inexact adders with different NAB values; NAB =4

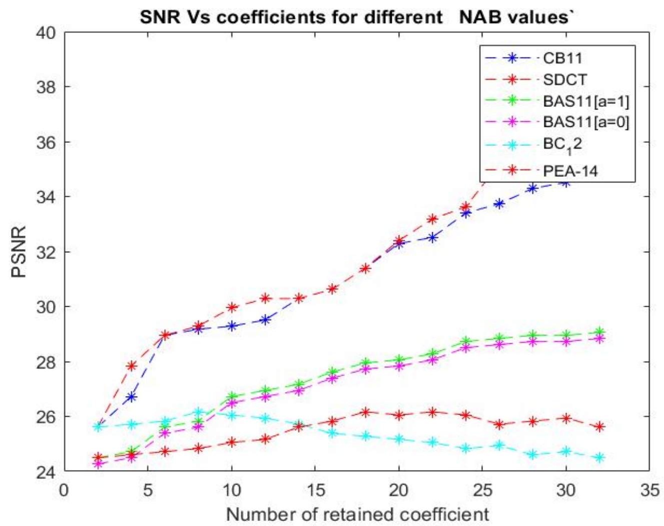

Fig5: Approximate DCT compression of an image using inexact adders with different NAB values; NAB =5

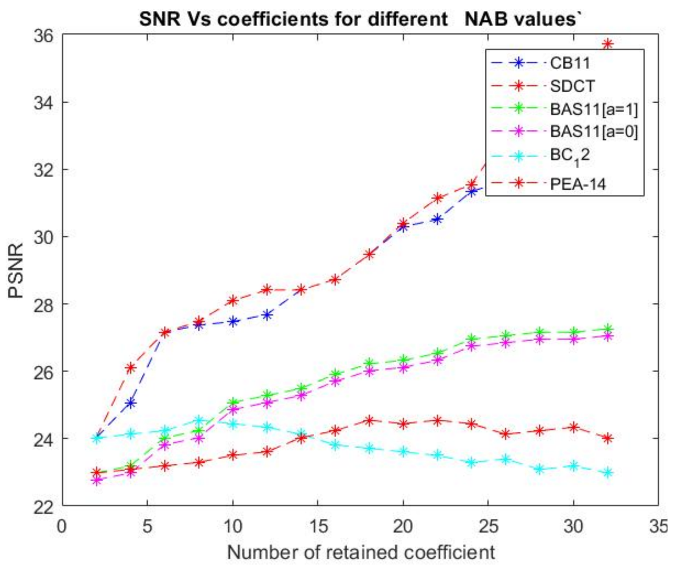

Fig6: Approximate DCT compression of an image using inexact adders with different NAB values; NAB =6

In Fig. 1). Fig. 2 shows that the MD between the uncompressed and inexact-compressed image pixels is reduced as more retained coefficients are used, the exceptions are PEA12 and BAS11 (a =2). This further confirms the previous results. Fig. 4 depicts the compressed Lena image using the most accurate CB11 method for three RC values, i.e., 4, 10 and 16 retained coefficients. This figure also shows for comparison purpose the exact DCT compression results with RC = 16. 4.3 Approximate DCT Using Inexact Computing Consider next the approximate DCT compression of Lena using inexact adders; as previously, the value of the NAB is increased from 3 to 6. The PSNR results are shown in Fig. 3,4,5,6 versus RC; the PSNR of the Compressed pictures (a measure of quality) are displayed by running all approximation DCT algorithms with just one inexact adder (for example, AMA1 as the inexact adder in the top row). Each column depicts the compressed picture quality by running all approximation DCT algorithms with just inexact adders (for a NAB value). The leftmost column, for example, is for NAB = 3. The PSNR degrades as the NAB grows, as predicted (an acceptable level of PSNR is obtained at a NAB value of 4). 4.4 Truncation Truncation is one of the inexact computing strategies that may be used; truncation outcomes are also displayed. The employment of inexact adders yields more precise results (truncation is performed at values of 3 and 4 bits).

5. CONCLUSION

This research introduced a novel method for compressing pictures by employing the Discrete Cosine Transform (DCT) algorithm. The suggested method consists of a three-level structure in which a multiplier-less DCT transformation (containing just adds and shift operations) is performed first, followed by high frequency component (coefficient) filtering and calculation utilizing inexact adders. It has been shown that by employing 8 x 8 picture blocks, each level contributes to an approximation in the compression process while still producing a very good quality image at the end. The findings of this publication suggest that the combined impacts of these three levels are well known; simulation and error analysis have shown a remarkable agreement in results for picture compression as an application of inexact computing.

Because the suggested framework has been shown to be successful for a DCT approach incorporating approximation at all three recommended levels, the following particular discoveries have been discovered and proven in this paper via simulation and analysis. analysis of errors When employing precise 16 bit

adders, CB11 delivers the greatest quality compression (highest PSNR values) of any approximation DCT approach (Fig. 1). Other quality indicators for image manipulation (AD and MD) PSNR findings were verified. Figures 1 and 2 Methods The next best approaches are BAS08, BAS11 with a = 0 and BAS11 with a = 1. InXA2 has been determined to perform the best among the inexact adders studied [14]. When using inexact adders. Non-truncation based approaches give superior results than related truncation schemes when implementing approximation DCT JPEG compression, particularly when considering greater NABs. (Figures 3, 4, and 6) When diverse pictures are utilized, the DCT calculated with inexact adders produces consistent results. In general, NAB values up to 4 provide sufficient compression. Then it was shown that using bigger NAB values reduces the quality of the findings significantly. Utilizing four picture benchmarks, the BC12 and PEA14 techniques require the least amount of execution time and energy to compress an image when compared to using an exact adder. When it comes to the greatest PSNR as a measure of picture quality, the approximate DCT technique CB11 delivers the highest value; nevertheless, when both execution time and energy savings are considered, the best approximate DCT method is BAS09.

REFERENCES

[1] R. E. Blahut, Fast Algorithms for Digital Signal Processing. Reading, MA, USA: Addison-Wesley, 1985

[2] V. Britanak, P. Yip and K. R. Rao, Discrete Cosine and Sine Trans forms. New York, NY, USA: Academic, 2007

[3] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for im- ages," IEICE Trans., vol. E-71, no. 11, pp. 1095–1097, Nov. 1988.

[4] D. Tun and S. Lee, "On the fixed-point-error analysis of several fast DCT algorithms," IEEE Trans. Circuits Syst. Vid. Technol., vol. 3, no. 1, pp. 27–41, Feb. 1993.

[5] C. Hsu and J. Yao, "Comparative performance of fast cosine trans form with fixed-point roundoff error analysis," IEEE Trans. Signal Process., vol. 42, no. 5, pp. 1256–1259, May 1994.

[6] J. Liang and T. D. Tran, "Fast multiplierless approximations of the fast DCT algorithms," IEEE Trans. Circuits Syst. Vid. Technol., vol. 3, no. 1, pp. 27–41, Feb. 1993.

[7] DCT with the lifting scheme," IEEE Trans. Signal Process., vol. 49, no. 12, pp. 3032–3044, Dec. 2001.

[8] Y. Dote and S.J. Ovaska, "Industrial applications of soft comput ing: A review," Proc. IEEE, vol. 89, no. 9, pp. 1243–1265, Sep. 2001.

[9] K. V. Palem, "Energy aware computing through probabilistic switching: A study of limits," IEEE Trans. Comput., vol. 54, no. 9, pp. 1123–1137, Sep. 2005.

[10] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proc. Des. Automat. Test Europe, 2010, pp. 957–960.

[11] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Lowpower digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[12] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 62, no. 9, pp. 1760–1771, Sep. 2013.

[13] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in Proc. ACM/IEEE Great Lakes Symp. VLSI, May 2015, pp. 343–348.

[14] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate xor/xnor-based adders for inexact computing," in Proc. IEEE Int. Conf. Nanotechnology, Aug. 2013, pp. 690—693.

[15] H. A. F. Almurib, T. Nandha Kumar and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in Proc. Conf. Des. Autom. Test Europe, Mar. 2016, pp. 660–665.

[16] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power DCT architecture," in Proc. Des. Automat. Test Europe, 2007, pp. 1–6.

[17] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, and N. Rajapaksha, "Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing," Meas. Sci. Technol., vol. 23, no. 11, pp. 1–15,Nov. 2012.

[18] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya, "Improved 8-point approximate DCT for image and video compression requiring only 14 additions," IEEE Trans. Circuits Syst. I Regular Papers, vol. 61, no. 6, pp. 1727–1740, Jun. 2014.

[19] N. Merhav and B. Vasudev, "A multiplication-free approximate algorithm for the inverse discrete cosine transform," in Proc. Int. Conf. Image Process., Oct. 24-28, 1999, pp. 759–763.

[20] C. Loeffler, A. Lightenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in Proc. IEEE Int. Conf. Acoustics Speech Signal Process., Feb. 1989, pp. 988–991.

[21] P. Duhamel and H. H'Mida, "New S DCT algorithms suitable for VLSI implementation," in Proc. IEEE Int. Conf. Acoustics Speech Signal Process., 1987, pp. 1805–1808.

[22] E. Feig and S. Winograd, "On the multiplicative complexity of discrete cosine transform," IEEE Trans. Inform. Theory, vol. 38, no. 4, pp. 1387–1391, Jul. 1992.

[23] V. Lecuire, L. Makkaoui, and J.-M. Moureaux, "Fast zonal DCT for energy conservation in wireless image sensor networks," Electron. Lett., vol. 48, no. 2, pp. 125–127, Jan. 19, 2012.

[24] T.I. Haweel, "A new square wave transform based on the DCT," Signal Process., vol. 82, pp. 2309–2319, 2001.

[25] K. Lengwehasatit and A. Ortega, "Scalable variable complexity approximate forward DCT," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 11, pp. 1236–1248, Nov. 2004.

[26] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Lowcomplexity 8 x 8 transform for image compression," Electron. Lett., vol. 44, no. 21, pp. 1249–1250, 2008.